

IES JOSÉ RODRIGO BOTET

PROYECTO DE ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

**SISTEMA DOMÓTICO CON UNA RASPBERRY PI**

|  |
| --- |
| NOMBRE: Germán Orlando Navarro Díaz  DNI: 50516754Z  NIA: 11395657  CURSO: 2016/2017  CONVOCATORIA: Ordinaria |

TRIBUNAL

Fdo. PRESIDENTE:

Fdo. TUTOR INDIVIDUAL:

Fdo.TUTOR COLECTIVO:

En Manises, a ……. de.......................................... de 2017.

|  |
| --- |
| NOTA FINAL: |
|  |

Contenido

[1. INTRODUCCIÓN 5](#_Toc484350167)

[2. ¿QUÉ ES LA RASPBERRY PI? 7](#_Toc484350168)

[2.1 ESPECIFICACIONES 8](#_Toc484350169)

[2.2 SoC (SYSTEM ON CHIP) 8](#_Toc484350170)

[2.3 CONEXIÓN INALÁMBRICA 9](#_Toc484350171)

[2.4 INTERFAZ GPIO 9](#_Toc484350172)

[2.4.1 ¿Qué se puede hacer con ellos? 10](#_Toc484350173)

[2.4.2 Corriente ​​ 10](#_Toc484350174)

[3. DESARROLLO DEL PROYECTO 12](#_Toc484350175)

[4. CONECTIVIDAD 14](#_Toc484350176)

[4.1 LAN (CABLE o INALÁMBRICO) 14](#_Toc484350177)

[4.2 DNS DINÁMICO. 14](#_Toc484350178)

[5. HARDWARE 16](#_Toc484350179)

[5.1 RELÉS 16](#_Toc484350180)

[5.2 SENSOR DE MOVIMIENTO PIR 16](#_Toc484350181)

[5.3 INTERRUPTOR MAGNÉTICO 18](#_Toc484350182)

[5.4 CÁMARA 19](#_Toc484350183)

[6. SOFTWARE 20](#_Toc484350184)

[6.1 SISTEMA OPERATIVO 20](#_Toc484350185)

[6.2 SERVIDOR WEB 20](#_Toc484350186)

[6.3 SSH, SECURE SHELL. 21](#_Toc484350187)

[7. MEJORAS DE FUTURO 22](#_Toc484350188)

[8. CONCLUSIONES 23](#_Toc484350189)

[9. BIBLIOGRAFIA 24](#_Toc484350190)

[ANEXO I Configuración servicio SSMTP 26](#_Toc484350191)

[ANEXO II Script para comprobar y enviar IP nuevo 27](#_Toc484350192)

[ANEXO III CÓDIGO FUENTE PÁGINA PRINCIPAL Y PYTHON 28](#_Toc484350193)

[ANEXO IV Cómo instalar el Sistema operativo. 39](#_Toc484350194)

[ANEXO V Reserva de dirección IP. 47](#_Toc484350195)

[ANEXO VI Instalación de php en el Raspberry Pi 49](#_Toc484350196)

[ANEXO VII Instalación de Apache en el Raspberry Pi 50](#_Toc484350197)

# INTRODUCCIÓN

Jack St. Claire Kilby, un nombre que probablemente nadie reconoce pero que gracias a él el mundo tecnológico es lo que es hoy en día. Esta persona inventó el circuito integrado. Esos pequeños componentes miniaturizados que se encuentran hoy en día en casi todos los dispositivos electrónicos que hacen que funcionen. Los encontramos en relojes, neveras, lavadoras, impresoras, coches, motores, proyectores, tamagotchis, y por supuesto en ordenadores, por mencionar algunos.

Gracias a la magia del cine y la ciencia-ficción muchas películas han cambiado nuestra forma de pensar con respecto a la tecnología. Nos han inspirado para crear y pensar que lo imposible tecnológicamente ahora es posible.

Todos estos años de investigación y curiosidad dieron paso a lo que hoy en día conocemos como domótica. La domótica es esa herramienta que nos permite automatizar, en nuestras casas o sitios de trabajo, sistemas para nuestra comodidad, por ejemplo, sistemas de climatización, encendido de luces a una determinada hora o ante la detección de movimiento, activar luces o aparatos a distancia y muchas más cosas.

¿Si es tan fabulosa la domótica por qué sólo un pequeño número de la población lo tiene/disfruta? Debido a sus costes de producción y la falta de compatibilidad entre marcas hace que cada uno establezca sus propios precios impidiendo el fomento de la competencia en el mercado tecnológico.

La fundación Raspberry Pi del Reino Unido es una ONG fundada en 2009 con el propósito de promover el estudio de la informática básica en escuelas. Diseñó, produjo, y comercializó la Raspberry Pi, una computadora miniatura de tamaño considerablemente inferior a los computadores que conocemos hoy en día. Este pequeño dispositivo nos permite desarrollar proyectos de domótica con un coste considerablemente más reducido ya que permite trabajar con módulos y piezas de distintos fabricantes con diferentes precios y calidades.

En la realización de este proyecto se presentan algunos de los mencionados accesorios o componentes con los que se puede interactuar. Se mostrarán relés, sensores de movimiento, imágenes capturadas por cámara, y un sensor magnético. Todo controlado a través de una interfaz web que se aloja en el Raspberry pi que también actúa como servidor web.

# ¿QUÉ ES LA RASPBERRY PI?

La Raspberry Pi es una computadora de bajo costo, del tamaño de una tarjeta de crédito que se conecta a un monitor o una televisión, y utiliza un teclado y un ratón estándar. Es un pequeño dispositivo capaz de permitir aprender a programar en lenguajes como Scratch y Python. Es capaz de hacer todo lo que se esperaría que una computadora de escritorio hiciera, desde navegar por Internet y reproducir video de alta definición, hasta hacer hojas de cálculo, procesamiento de textos y juegos. Además, permite la conexión directa de accesorios, aparatos, sensores, o dispositivos que pueden ser controlados conectándolos de modo remoto.

|  |  |
| --- | --- |
| ***Raspberry Pi 3***  ***Vista superior*** | ***Raspberry Pi 3***  ***Vista inferior*** |
| ***Raspberry Pi 3Vista general*** |  |

Para la realización de este proyecto se utilizó la Raspberry pi modelo 3, actualmente el modelo más reciente que se comercializa en este tamaño.

## ESPECIFICACIONES

|  |  |
| --- | --- |
| SoC (System On Chip) Broadcom BCM2837 | |
| CPU: | 1.2GHz 64-bit quad-core ARMv8 |
| GPU: | Broadcom VideoCore IV |
| RAM: | 1GB LPDDR2 (900 MHz) |
| Red: | Ethernet:10/100,  Inalámbrica:2.4GHz 802.11n |
| Bluetooth: | Bluetooth 4.1 Classic,  Bluetooth de baja energía (BLE) |
| Almacenamiento: | microSD,  USB |
| GPIO: | 40 pines |
| Puertos: | HDMI,  conector de audio y video analógico de 3,5 mm,  4× USB 2.0,  Ethernet,  Interfaz Serial de Cámara (CSI),  Interfaz Serial de Pantalla (DSI) |

## SoC (SYSTEM ON CHIP)

Construido específicamente para el nuevo Pi 3, el sistema de Broadcom BCM2837 (SoC) incluye cuatro núcleos de procesamiento ARM Cortex-A53 de alto rendimiento que funcionan a 1.2GHz con una memoria caché de Nivel 2 de 32kB y 512kB de Nivel 1, un procesador gráfico VideoCore IV, y está conectado a un módulo de memoria LPDDR2 de 1 GB en la parte posterior de la placa.

## CONEXIÓN INALÁMBRICA

|  |
| --- |
|  |
| *Antena Wi-fi y Bluetooth* |

El chip Broadcom BCM43438, proporciona una red inalámbrica 802.11n de 2.4GHz, Bluetooth Low Energy y Bluetooth 4.1 Classic. Inteligentemente construido directamente en la placa para mantener los costes bajos.

Sus radios están conectados a esta antena chip soldada directamente a la placa, con el fin de mantener el tamaño del dispositivo a un mínimo. A pesar de su reducido tamaño, esta antena debe ser más que capaz de captar las señales inalámbricas LAN y Bluetooth.

## INTERFAZ GPIO

|  |
| --- |
|  |
| *Interfaz GPIO* |

GPIO: **G**eneral **P**urpose **I**nput **O**utput

Sería algo así como "interfaz general de entrada y salida'. Una de las características más poderosas de la Raspberry pi. Son esos 40 pines (26 pines en la Raspberry1 y 2), que son la interfaz física entre la Raspberry pi y el mundo exterior. Un modo de entender su funcionamiento es pensar en ellos como interruptores que se pueden activar o desactivar.

### ¿Qué se puede hacer con ellos?

​​La distribución numérica del GPIO puede ser BCM, que corresponde con el orden asignado por el fabricante y no tiene un orden lógico aparente; o numeración GPIO física, es el orden que tienen los pines físicamente en la placa.

Los pines se pueden programar para interactuar con el mundo real. **Input** (entrada) La interacción entrante puede ser información recibida por un sensor o una señal de otro ordenador o dispositivo, por ejemplo, cuando recibe la temperatura de sensores térmicos, o la señal que indica que una puerta se ha abierto, etc. Cuando se envía información de entrada como señal (3,3 V) a la Raspberry Pi el pin puede confundir su estado actual con encendido o apagado, por lo que se dice que el pin tiene un estado flotante sin una referencia. Por eso cuando se configura una entrada se debe usar un "pull-up" o "pull-down” para poder definir la señal correctamente y tener la certeza de qué tipo es la entrada

Lo mismo sucede con la información de salida, **output**, se puede activar un LED, enviar una señal o información a otro dispositivo. Si la Raspberry pi está conectada a una red (Internet o LAN) se pueden controlar dispositivos que estén conectados a esa misma red sin importar su ubicación, estos mismos dispositivos pueden a la vez enviar información de vuelta. Cuando un pin se configura como output tiene dos opciones: estar en estado HIGH (True o 1) o LOW (False o 0). HIGH significa que emite 3.3 voltios, y LOW que está apagado.

### Corriente ​​

La Raspberry Pi también posee varios pines con salidas de 5v, 3.3v y tierra (GND) a los cuales se puede conectar el hardware que se desee siempre que el amperaje usado no sea superior a 50mA para los pines de 3.3v, y alrededor de 700mA para los de 5v . Hay que tener cuidado cuando se conectan los pines al hardware externo, ya que la Raspberry Pi carece de ciertas protecciones electrónicas que si se conecta mal uno de los pines de corriente se podría generar un corto circuito y dejar inservible la Raspberry.

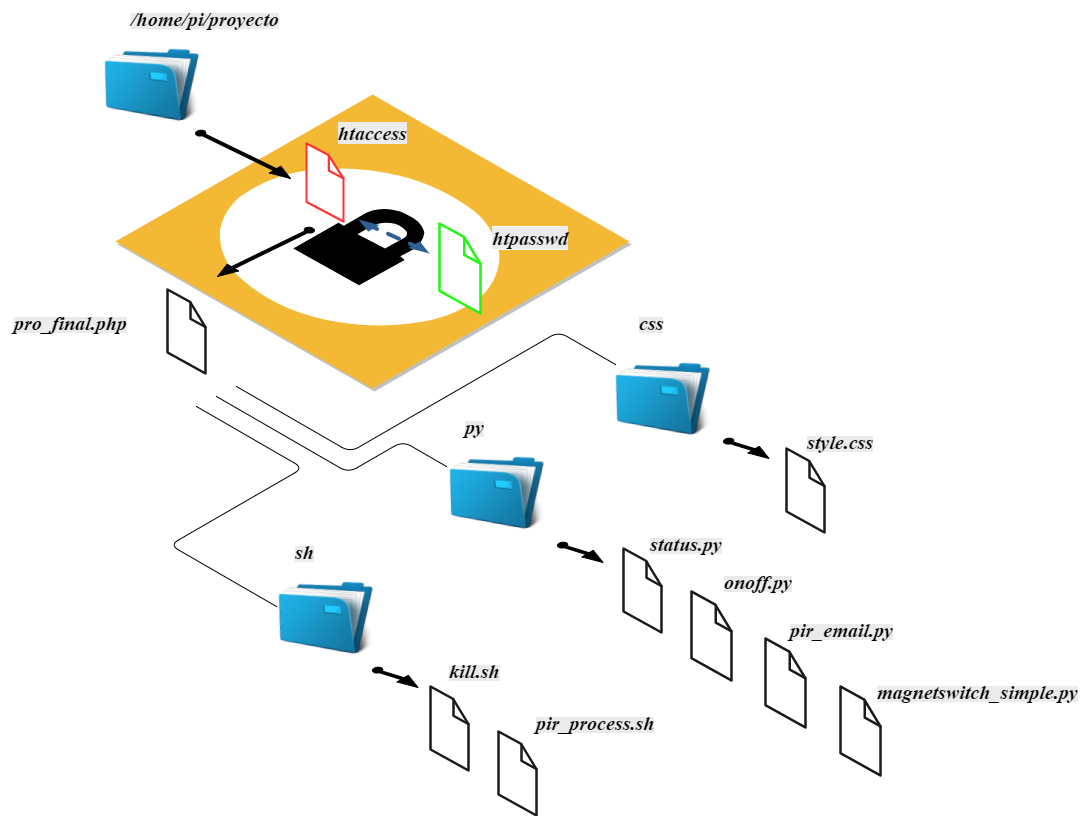
# DESARROLLO DEL PROYECTO

En este proyecto de domótica se usan varios componentes y accesorios he hardware que usan como base el mini ordenador conocido como Raspberry pi. La razón para escoger este dispositivo como eje central del proyecto es la facilidad y versatilidad de los usos que se le pueden dar. Como ejemplo se presentan varios dispositivos/accesorios tales como un **módulo de relés**, un sensor infrarrojo pasivo (**PIR**), un **sensor/interruptor magnético**, y la cámara diseñada de modo exclusivo para la Raspberry (**Raspicam**), todos ellos controlados a través de una página web creada con **HTML**, **PHP**, **Python**, y **Shell Script.**

El objetivo principal es que todo lo que se diseñó e implementó sea controlado desde cualquier ubicación del mundo usando como herramienta cualquier dispositivo que posea un navegador web, por ejemplo, un teléfono móvil, una Tablet, un ordenador o incluso un reloj de pulso (smartwatch).

|  |
| --- |
|  |
| *Esquema de funcionamiento* |

El sistema funciona usando como interfaz una página web creada con HTML, y PHP. A través de PHP se ejecutan comandos en la Raspberry Pi en Python y Shell script para activar o desactivar el hardware conectado, para enviar correos electrónicos cuando suceda un evento iniciado por uno de los sensores, o para indicar el estado de dichos sensores.



# CONECTIVIDAD

## LAN (CABLE o INALÁMBRICO)

Este es el modo más eficiente y efectivo a la hora de conectarse a un dispositivo de red. En este proyecto se usó una conexión por cable de tipo UTP-5e para la conexión entre la Raspberry pi y el router. También sería posible realizar la conexión a la Raspberry Pi de modo inalámbrico según se prefiera.

## DNS DINÁMICO.

Una de las formas de conectarse a la Raspberry Pi es a través de Internet. Para ello, si no tenemos una **IP estática pública** de nuestro proveedor de internet, se hace necesario usar un servicio de **DNS dinámico** para que éste nos provea nuestra dirección IP pública y así poder establecer una conexión desde afuera de la LAN a la Raspberry. Con el fin de evitar la dependencia de dicho servicio externo se usa una aplicación **SSMTP**, **ANEXO1**, que nos envía nuestra dirección IP pública al e-mail que se le indique. Para ello se creó un script, **ANEXO II**, el cual usa Crontab (el programador de tareas de Linux) para que cada 3 minutos revise si nuestra dirección IP ha cambiado. Este email es enviado desde la Raspberry Pi de modo automático con la dirección IP pública actual **sólo** cuando ésta ha cambiado. De este modo no dependemos de ningún servicio externo más que del Internet mismo.

|  |
| --- |
|  |
| *Script que envía nueva dirección IP* |

# HARDWARE

## RELÉS

|  |
| --- |
| Related image |
| *Relé* |

Un relé **es un interruptor** accionado por un electroimán. Para funcionar se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

## SENSOR DE MOVIMIENTO PIR

|  |
| --- |
| proximity_pirsensor.jpg |
| *Sensor PIR* |

(Passive InfraRed, Infrarrojo Pasivo)

Los sensores PIR permiten detectan movimiento, casi siempre se usan para detectar si un humano se ha movido dentro o fuera de un espectro. Son pequeños, económicos, de bajo consumo, fáciles de usar y no se desgastan. Por esa razón que se encuentran comúnmente en los aparatos y aparatos utilizados en los hogares o empresas. A menudo se les conoce como sensores PIR, "infrarrojos pasivos", o “piroeléctrico".

|  |
| --- |
| proximity_pyrosensor.gif |
| *Sensor PIR* |

Los PIRs están hechos de un sensor piroeléctrico que puede detectar niveles de radiación infrarroja. Todo emite una radiación de bajo nivel, y cuanto más caliente es, más radiación se emite. El sensor PIR tiene dos mitades, cada mitad está hecha de un material especial que es sensible a IR. Cuando el sensor está inactivo, ambas mitades detectan la misma cantidad de IR, la cantidad ambiental irradiada desde la habitación o las paredes o al aire libre. Cuando pasa un cuerpo caliente como un ser humano o un animal, primero intercepta una mitad del sensor PIR, lo que causa un cambio diferencial positivo entre las dos mitades. Cuando el cuerpo caliente sale del área de detección, sucede lo contrario, por lo que el sensor genera un cambio diferencial negativo. Estos impulsos de cambio son lo que se detecta.

|  |
| --- |
| proximity_linearfresnel.gif |
| *Lente Fresnel* |

La lente Fresnel que se coloca encima del sensor condensa toda la radiación infrarroja de alrededor en un espacio mucho más pequeño, esto le permite abarcar al sensor mucho más espacio, tal como sucede con el sensor de una cámara fotográfica para abarcar una imagen más grande.

|  |
| --- |
| proximity_frenelling.jpg |
| *Vista interior lente Fresnel* |

Visto desde el interior las diferentes facetas y sub-lentes crean una gama de áreas de detección, intercaladas entre sí. Es por eso que los centros de las lentes en las facetas de arriba son 'inconsistentes' - cada uno apunta a una mitad diferente del elemento de detección de PIR

## INTERRUPTOR MAGNÉTICO

|  |
| --- |
| Image result for magnet switch |
| *Interruptor magnético* |

Este sensor es esencialmente un interruptor de lámina contenido en una envoltura de plástico. Normalmente la lámina hace que esté “abierto" (no hay conexión entre los dos cables). La otra mitad es un imán. Cuando el imán está a menos de 13 mm de distancia, el interruptor de láminas se cierra y se utilizan a menudo para detectar cuando una puerta o un cajón está abierto, por lo que tienen pestañas de montaje y tornillos.

## CÁMARA

|  |
| --- |
| Image result for raspicam |
| *Cámara (Raspicam)* |

El módulo de la cámara de la Raspberry Pi es un complemento diseñado para ella, denominada **Raspicam**. Se conecta por medio de uno de los dos pequeños conectores en la superficie superior. Esta interfaz utiliza la interfaz dedicada de CSI, que fue diseñada especialmente para la interfaz a las cámaras. El bus CSI (Interfaz Serial de la Cámara) es capaz de velocidades de datos extremadamente altas, y lleva exclusivamente datos de píxeles.

Es pequeña, en torno a 25 mm x 20 mm x 9 m. También pesa poco más de 3 g, lo que lo hace perfecta para móviles u otras aplicaciones donde el tamaño y el peso son importantes. Se conecta a través de un cable de cinta corta. La cámara está conectada al procesador BCM2835 en el Pi a través del bus CSI, un enlace de mayor ancho de banda que transporta los datos de píxeles de la cámara al procesador. Este bus viaja a lo largo del cable plano que une la placa de la cámara al Pi.

El sensor en sí tiene una resolución nativa de 5 megapíxeles, y tiene una lente de enfoque fijo. En términos de imágenes fijas, la cámara es capaz de imágenes estáticas de 2592 x 1944 píxeles, y también soporta 1080p30, 720p60 y 640x480p60 / 90 de vídeo.

# SOFTWARE

## SISTEMA OPERATIVO

El sistema operativo oficial basado en Linux diseñado para la Raspberry es Raspbian Jessie. En el Anexo VI se explica el proceso de instalación detalladamente. La versión instalada ha sido:

***Raspbian Jessie con PIXEL***

*Imagen con Escritorio PIXEL basado en Debian Jessie*

*Version:* ***April 2017***

*Fecha de Lanzamiento: 10 abril* ***2017***

*Version del Kernel:* ***4.4***

## SERVIDOR WEB

Apache/2.4.10 Es el servidor web instalado para la realización de este proyecto. En la versión de abril 2017 de Raspbian ya viene instalado por defecto. En el **ANEXO VII** se explica cómo instalarlo.

El servidor web funciona como plataforma principal para desarrollar la interfaz gráfica, ya que los controles del hardware se diseñaron y construyeron para activarlos, desactivarlos, y controlar su estado a través de una página web. Esto permite poder acceder a los controles desde cualquier dispositivo que tenga instalado un navegador web tales como un ordenador, teléfono móvil, Tablet, reloj (smartwatch), etc.

Los lenguajes de desarrollo empleados en la construcción de la página web para este proyecto fueron HTML y PHP; y para la manipulación del hardware conectado a la Raspberry Pi fueron Python y Shell Script.

## SSH, SECURE SHELL.

Es un protocolo y el nombre del programa que lo implementa, sirve para acceder de modo seguro a máquinas a través de redes no seguras en una arquitectura cliente servidor. Permite el control total de una máquina a través de la línea de comandos. Se creó para reemplazar Telnet, el cual enviaba información como contraseñas o usuarios como texto plano.

# MEJORAS DE FUTURO

Entre alguna de las mejoras que se podrían aplicar a este proyecto se podrían destacar:

* La utilización de una base de datos para registrar cada evento que sucede ya que en la Raspberry también se pueden instalar algún motor de base de datos, por ejemplo, MySQL.
* El uso de Javascript permitiría visualizar en tiempo real el cambio de estado de los pines GPIO.
* Las imágenes capturadas aparte de enviarlas por email se podrían almacenar en un servidor FTP, o en servicios en la nube como Google Drive, OneDrive, o Dropbox.
* Las tareas o rutinas se pueden programar para que se ejecuten con una determinada frecuencia en cualquier cantidad de tiempo, desde milisegundos a años.

# CONCLUSIONES

Después de haber llevado a cabo este proyecto se podría concluir que existen numerosas posibilidades a la hora de construir y diseñar elementos de automatización para el día a día. La facilidad para adquirir componentes hardware, tales como sensores, relés, motores, y otra gran cantidad de elementos es muy asequible. Además de la facilidad para encontrar el hardware, el software base y las herramientas de software son 100% libres y gratuitas, sin embargo, se debe tener el conocimiento para programar en el lenguaje necesario para manipular el hardware, por ejemplo, Python o JavaScript; sumado a esto existen los foros en Internet y los tutoriales que son de gran ayuda para aclarar dudas o realizar proyectos de pequeña envergadura.

# BIBLIOGRAFIA

* The MagPi Magazine. (2017). *Raspberry Pi 3 is out now! Specs, benchmarks & more - The MagPi Magazine*. [online] Disponible en: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/ [Acceso Marzo. 2017].
* (2017). *Raspbery Pi – Control de la cámara con Python*. [online] Disponible en: http://fpaez.com/raspbery-pi-control-de-la-camara-con-python/ [Accessed 4 Jun. 2017].
* +3.3v?, G. (2017). *GPIO: Why wire button to ground rather than +3.3v?*. [online] Raspberrypi.stackexchange.com. Disponible en: https://raspberrypi.stackexchange.com/questions/9481/gpio-why-wire-button-to-ground-rather-than-3-3v [Acceso Marzo. 2017].
* GPIO.PUD\_DOWN?, C. (2017). *Change the GPIO pull\_up\_down=GPIO.PUD\_UP to GPIO.PUD\_DOWN?*. [online] Raspberrypi.stackexchange.com. Disponible en: https://raspberrypi.stackexchange.com/questions/27263/change-the-gpio-pull-up-down-gpio-pud-up-to-gpio-pud-down [Acceso Marzo. 2017].
* Resistorguide.com. (2017). *Pull up resistor / Pull down resistor » Resistor Guide*. [online] Disponible en: http://www.resistorguide.com/pull-up-resistor\_pull-down-resistor/ Acceso Marzo. 2017].
* Abarca, A. (2017). *Usar un sensor de puerta magnetico en un Raspberry Pi*. [online] Blog de Alexis Abarca. Disponible en: https://www.alexisabarca.com/2016/01/usar-un-sensor-de-puerta-magnetico-en-un-raspberry-pi/ [Acceso Marzo. 2017].
* Learn.adafruit.com. (2017). *Overview | PIR Motion Sensor | Adafruit Learning System*. [online] Disponible en: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview [Acceso Marzo. 2017].

1. Configuración servicio SSMTP

Se debe configurar en Gmail el inicio de sesión con la verificación en dos pasos y también generar una contraseña de aplicación.

Una vez se tenga Gmail configurado y se haya conseguido una "contraseña\_de\_aplicación\_de\_google" se ha de instalar lo siguiente:

sudo apt-get install ssmtp mailutils

Hay que editar el archivo de configuración del servicio SSMTP

sudo nano /etc/ssmtp/ssmtp.conf

Hay que copiar el siguiente contenido en el archivo ajustándolo a la información necesaria.

root=usuario\_gmail

mailhub=smtp.gmail.com:587

rewriteDomain=gmail.com

hostname=gmail.com

AuthUser=usuario\_gmail@gmail.com

AuthPass=contraseña\_de\_aplicacion\_de\_google

FromLineOverride=YES

UseSTARTTLS=YES

Lo siguiente es programar una tarea para que compruebe cada cierto tiempo si el IP ha cambiado. Para esto se usa "crontab" el programador de tareas de Linux.

Acceder a crontab

sudo crontab -u pi -e

\*/3 \* \* \* \* /home/pi/ip.sh

El "\*/3" en la línea anterior indica que la tarea se realizará cada 3 minutos

1. Script para comprobar y enviar IP nuevo

#!/bin/bash

#ip.sh

#comprobacion de existencia del archivo que contiene el ip

if [ -f "ip" ]; then

#busqueda del ip actual

ip1=`wget http://ipinfo.io/ip -qO -`

#lectura del ultimo ip para comparar con el actual

ip2=`catip`

#comparacion de los últimos dos ips y longitud de cadena

if [ "$ip1" != "$ip2" ] && [ `cat ip | grep -x ‘.\{3,16\}’`];then

#envio de la direccionip por email

echo http://"$ip1" | mail -s "IP CHANGE" usuario@gmail.com

fi

else

#creación del archivo ip en caso de que no exista

echo "@" >ip

fi

1. CÓDIGO FUENTE PÁGINA PRINCIPAL Y PYTHON

PÁGINA PRINCIPAL: pro\_final.php

<?php session\_start();

//lectura de los pines usados para definir el color de fondo

$bg = array(26,25,24,23,22,21);

foreach($bg as $value){

if(!isset($\_SESSION[$value])){ //se usan sesiones para no tener que consultar cada vez que se cargue la página, ahorra energía.

$\_SESSION[$value]=read($value); //llamada a la función de lectura de estado de pines

if ($\_SESSION[$value]==0){ $bg[$value]="button\_on";}else{$bg[$value]="button\_off";}

}

}

/\*FUNCION PARA LEER EL ESTADO DE LOS PINES\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

function read($pinn){

$state=exec("sudo python status.py $pinn");

return $state;

}

/\*FIN DE FUNCIÓN\*/

/\*FUNCION PARA ASIGNAR COLOR DE FONDO\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

function bgcolor($pin){

$\_SESSION[$pin]=read($pin);

if ($\_SESSION[$pin]==0){

$bg[$pin]="button\_on";

}else{

$bg[$pin]="button\_off";

}

return $bg[$pin];

}

/\*FIN DE FUNCIÓN\*/

/\* PHP PIR\_EMAIL \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(isset($\_POST['pir\_email'])){

$pin=$\_REQUEST['pir\_email'];

exec('sudo python pir\_email.py 17 > /dev/null 2>/dev/null &');

}/\* FIN DE PIR\_EMAIL \*/

/\* PHP BOTON OFF PIR \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(isset($\_POST['off\_pir'])){

exec("sudokillallpython");

}/\* FIN \*/

/\* PHP PIR STATE \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$state17=exec("if [ `ps -C python | wc -l` -gt 1 ]; then echo 1; fi");

if ($state17>0){

$bgpir="pir\_running";

}else{

$bgpir="pir\_not\_running";

}/\* FIN \*/

/\* PHP BOTON SWITCH \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(isset($\_POST['switch'])){

$pin=$\_REQUEST['switch'];

exec("sudo python onoff.py $pin");

$bg[$pin]=bgcolor($pin);

}/\* FIN DE BOTON SWITCH \*/

/\* PHP SENSOR PUERTA \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$state26=read(26); //llamada a la función de lectura de estado de pines

if ($state26==0){

$bga[26]="openclose\_on";

}else{

$bga[26]="openclose\_off";

}/\* FIN \*/

/\* PHP OFF \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(isset($\_POST['off'])){

exec("sudo shutdown -h now");

}/\* FIN \*/

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="author" content="German Navarro">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Proyecto final ASIR 2017</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<div class="contenedora">

<div class="bloquea">

<H1>

ACTIVAR DISPOSITIVOS

</H1>

<form method="post" action="<?php $\_SERVER['PHP\_SELF']?>">

<div class="tablabord pad1">

<div class="">

<div class="">

Sensor de movimiento + cámara

</div>

<div class=""><button class="" type="submit" name="pir\_email" value="17">On</button></div>

<div class=""><button class="" type="submit" name="off\_pir" >OFF</button></div>

<div class="<?php echo $bgpir; ?>"></div>

</div>

</div>

</form>

<div class="tablabord pad1">

<div class="">

<div>

<form method="post" action="<?php $\_SERVER['PHP\_SELF']?>">

<button class="" type="submit" name="reload" id="puerta" >CHECK</button>

</form>

</div>

<div class="">Puerta: </div>

<div class="">

<div class="<?php echo $bga[26];?>"></div>

</div></div></div>

<br/>

<form method="post" action="<?php $\_SERVER['PHP\_SELF']?>">

<div class="tablabord pad1">

<div class=""><div class="">

Compuerta (DELETE PENDING)

<button class="<?php echo $bg[25] ?>" type="submit" name="stepper" value="25">ON / OFF</button>

</div></div></div>

</form>

<form method="post" action="<?php $\_SERVER['PHP\_SELF']?>">

<div class="tablabord pad1">

<div class="">

<div class="">Puerta</div>

<div class="">

<button class="<?php echo $bg[24] ?>" type="submit" name="switch" id="puerta" value="24">ON / OFF</button>

</div></div>

<div class="">

<div class="">Luz pasillo</div>

<div class="">

<button class="<?php echo $bg[23] ?>" type="submit" name="switch" id="luzpasillo" value="23">ON / OFF</button>

</div>

</div>

<div class="">

<div class="">Ventilador</div>

<div class="">

<button class="<?php echo $bg[22] ?>" type="submit" name="switch" id="ventilador" value="22">ON / OFF</button>

</div>

</div>

<div class=""><div class="">Aula </div><div class="">

<button class="<?php echo $bg[21] ?>" type="submit" name="switch" id="aula" value="21">ON / OFF</button>

</div></div></div>

</form>

<br/>

<form method="post" action="<?php $\_SERVER['PHP\_SELF']?>">

APAGAR RASPBERRY

<button type="submit" name="off" value="">SHUTDOWN</button>

</form>

</div>

</div>

</body>

</html>

***style.css***

body {

background: #C5CAE9;

color: #000;

font-weight:800;

font-family: helvetica, arial, sans-serif;

padding:none;

}

h1 {text-shadow: 2px 2px2px grey, 2px 2px2pxlightblue;}

.contenedora {

max-width: auto;

padding:none;

}

.bloquea {

box-shadow: 10px 10px 5px grey ;

font-family: 'Open Sans', sans-serif;

max-width: 700px;

max-height:100%;

margin: 0 auto;

padding:1em;

background: #F0F4C3;

border-bottom-left-radius:20px ;

border-bottom-right-radius:20px ;

}

.bord {

border: 0.5px solid lightgrey;

border-top-left-radius:14px;

border-bottom-right-radius:14px;

}

.pad1{

padding:12px;

}

div.tabla {

display: table;

width:auto;

}

div.tabla> div {

display: table-row;

}

div.tabla> div > div {

display: table-cell;

border-bottom: 1px solid black;

padding:7px 9px;

}

.button\_on{

display: inline-block;

padding: 5px 20px;

font-size: 18px;

cursor: pointer;

text-align: center;

outline: none;

color: #fff;

background-color: #008CBA;

border: none;

border-radius: 8px;

box-shadow: 0 6px #999;

text-shadow: 2px 2px2px grey, 2px 2px2px pink;

}

.button\_on:hover {background-color: #22aedd}

.button\_on:active {

background-color: #444444;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

.button\_off{

display: inline-block;

padding: 5px 20px;

font-size: 18px;

cursor: pointer;

text-align: center;

outline: none;

color: #fff;

background-color: #444444;

border: none;

border-radius: 8px;

box-shadow: 0 6px #999;

text-shadow: 2px 2px2px grey, 2px 2px2px pink;

}

.button\_off:hover {background-color: #aaaaaa}

.button\_off:active {

background-color: #666666;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

.openclose\_on:before{

content:'OPEN';

display: inline-block;

padding: 5px 20px;

font-size: 18px;

// cursor: pointer;

text-align: center;

outline: none;

color: #fff;

background-color: #008CBA;

border: none;

border-radius: 8px;

box-shadow: 0 6px #999;

text-shadow: 2px 2px2px grey, 2px 2px2px red;

}

.openclose\_off:before{

content:'CLOSED';

display: inline-block;

padding: 5px 20px;

font-size: 18px;

// cursor: pointer;

text-align: center;

outline: none;

color: #fff;

background-color: #444444;

border: none;

border-radius: 8px;

box-shadow: 0 6px #999;

text-shadow: 2px 2px2px grey, 2px 2px2px white;

}

.pir\_running:before{

content:'RUNNING';}

.pir\_not\_running:before{

content:'NOT RUNNING';}

**CÓDIGO PYTHON:**

***status.py***

#!/usr/bin/python

#este script recibe un numero de pin por parametro y devuelve su estado

import RPi.GPIO as GPIO

import sys

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#recibir el pin como parametro

pin=(int(sys.argv[1]))

#GPIO.setup(pin, GPIO.OUT)

GPIO.setup(pin, GPIO.OUT)

status=GPIO.input(pin)

print status

***pir\_email.py***

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

import picamera

from datetime import datetime

from email.mime.text import MIMEText

from email.mime.application import MIMEApplication

from email.mime.multipart import MIMEMultipart

from smtplib import SMTP

import smtplib

import sys

DEST='microkom@gmail.com'

OWNER='microkom02@gmail.com'

PIR=(int(sys.argv[1]))

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(PIR, GPIO.IN)

current\_state = 0

previous\_state = 0

try:

print "Esperando a que el sensor se ajuste al espacio.."

#loop hasta que el PIR output sea 0

while GPIO.input(PIR)==1:

current\_state = 0

print " Listo"

#Loop hasta que el usuario presione CTRL + C

while True:

#comprobar el estado del sensor

current\_state = GPIO.input(PIR)

if current\_state == 1 and previous\_state==0:

#Sensor activado

print" Movimiento detectado"

hora= datetime.now().strftime('%Y%m%d\_%H%M%S')

nombre= "foto\_%s.jpg" %hora

time.sleep(1)

#captura de la foto

try:

camera = picamera.PiCamera()

#camera.rotation = 270

camera.capture(str(nombre))

finally:

camera.close()

#envio por email de la foto capturada

recipients = [DEST]

emaillist = [elem.strip().split(',') for elem in recipients]

msg = MIMEMultipart()

msg['Subject'] = str(nombre)

msg['From'] = OWNER

msg['Reply-to'] = OWNER

msg.preamble = 'Multipart massage.\n'

part = MIMEText("Fotocapturada")

msg.attach(part)

part = MIMEApplication(open(str(nombre),"rb").read())

part.add\_header('Content-Disposition', 'attachment', filename=str(nombre))

msg.attach(part)

server = smtplib.SMTP("smtp.gmail.com:587")

server.ehlo()

server.starttls()

server.login(OWNER, "contraseña\_de\_aplicacion\_de\_google")

server.sendmail(msg['From'], emaillist , msg.as\_string())

#fin de envio

#att3.attachment(nombre)

#grabar estado anterior

previous\_state=1

elifcurrent\_state==0 and previous\_state==1:

#sensor ha vuelto a su estado de reposo

print " Listo"

previous\_state=0

#esperar 10 milisegundos

time.sleep(0.1)

except KeyboardInterrupt:

print " Fin"

#reset GPIOs

GPIO.cleanup()

***Onoff.py***

#!/usr/bin/python

#este script recibe un numero de pin por parametro y lo enciende o apaga

#segun sea su estado

importRPi.GPIO as GPIO

import sys

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#recibir el pin como parametro

pin=(int(sys.argv[1]))

GPIO.setup(pin, GPIO.OUT)

#lectura del estado del pin

status=GPIO.input(pin)

#encendido y apagado del pin

if status==0:

GPIO.output(pin, GPIO.HIGH)

else:

GPIO.output(pin, GPIO.LOW)

***magnetswitch\_simple.py***

#!/usr/bin/python

import RPi.GPIO as GPIO

import sys

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

pin=(int(sys.argv[1]))

GPIO.setup(pin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

try:

if GPIO.input(pin)==True:

print "1"

else:

print "0"

except KeyboardInterrupt:

GPIO.cleanup()

finally:

GPIO.cleanup()

**SCRIPT SHELL**

***pir\_process.sh***

#!/bin/bash

#lee si el proceso pir\_email.py está activo y devuelve su estado

if [ -z `/bin/ps -fu $USER |grep "pir\_email.py" | grep -v "grep"|cut -d" " -f 21` ]; then

echo 1

else

echo 0

fi

***kill.sh***

#!/bin/bash

#termina específicamente el proceso pir\_email.py

valor=`/bin/ps -fu $USER |grep "pir\_email.py" |grep -v "grep" |awk '{print $2}'`

sudo kill -9 $valor

1. Cómo instalar el Sistema operativo.

La instalación del sistema operativo Raspbian, en este caso, se realizó sin conectar la Raspberry a un monitor, se accedió a ella desde otro ordenador a través de SSH.

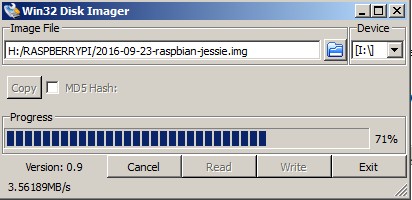
Se necesitó:

* Hardware
  + Raspberry Pi con su fuente de alimentación
  + Tarjeta MicroSD (y lector de MicroSD si hiciese falta)
  + Cable de red con conectores RJ45 (necesario para conectarlo al router o switch de casa)
  + Ordenador / Computador
* Software
  + Imagen del sistema operativo (es el software base sobre lo que funcionan todas las aplicaciones). Descargar la imagen de Raspbian de la página [www.raspberrypi.org](https://www.raspberrypi.org/downloads/raspbian/)
  + SDformatter para formatear la tarjeta SD. [www.sdcard.org](https://www.sdcard.org/downloads/index.html)
  + Descarga Win32DiskImager desde <https://sourceforge.net/projects/win32diskimager/files/Archive/> para copiar la imagen en la tarjeta.
  + “AdvancedIp scanner”  <http://www.advanced-ip-scanner.com/> para averiguar el IP de la Raspberry
  + Putty <http://www.putty.org/> para conectarnos vía SSH al Raspberry

Una vez descargado todo el software lo primero que hay que hacer es:

1. Formatear la tarjeta microSD usando SDformatter.
2. Escribe la imagen que se descargó de Raspbian en la tarjeta microSD

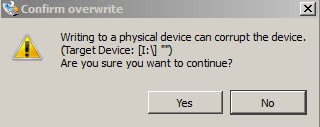
Localizar el archivo imagen y seleccionar la letra que tiene asignada la tarjeta microSD



Letra que tiene asignada la tarjeta microSD

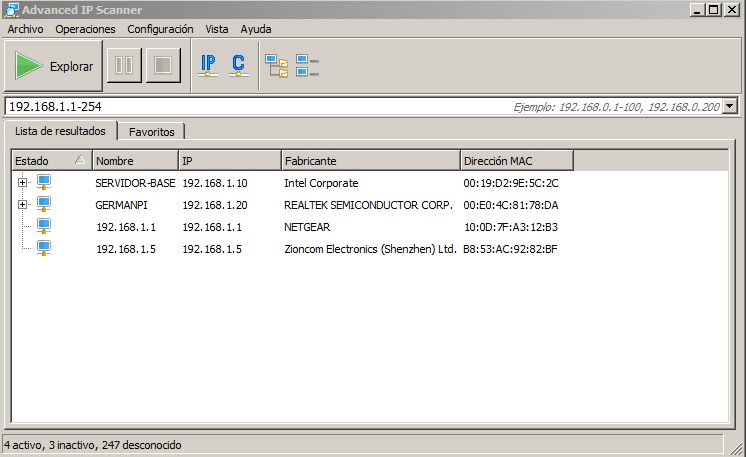
Ubicación y nombre de archivo imagen

Clic en "Yes" para continuar



Advertencia de seguridad de perdida de datos

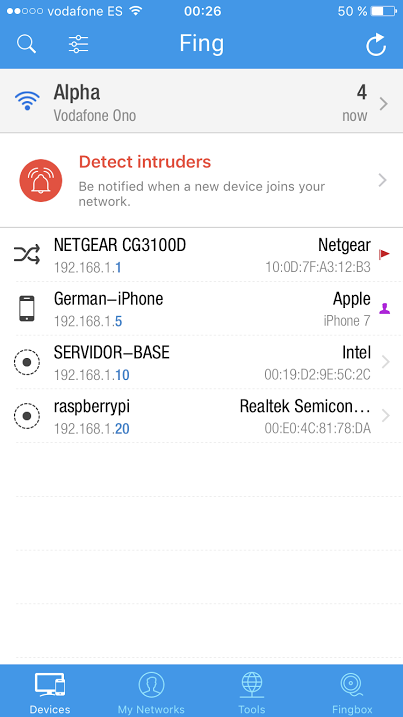
1. Una vez finalizada la copia de la imagen en la microSD, y debido a la última actualización de seguridad del 2017, **hay que crear un archivo con el nombre "ssh" en la partición boot de la tarjeta para poder acceder por SSH**a la Raspberry. Inserta la tarjeta en el Raspberry, conecta el cable de red al router de casa y al Raspberry. Conecta el cable de alimentación. Espera alrededor de 3 minutos hasta que el sistema operativo se ejecute por primera vez. El archivo "ssh" que creaste antes se borra automáticamente, lo que significa que debes acceder a [Configuración avanzada](http://aprendoycomparto.es/raspberry.html#config), Interfacing options, para activar el servicio SSH.
2. Ejecuta Advanced IP scanner y averigua la dirección IP del Raspberry. También puedes usar la aplicación "Fing" disponible para Android y Apple.



Dirección IP Raspberry con nombre de Host “GERMANPI”

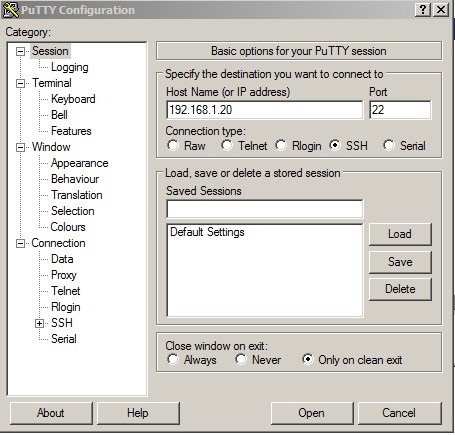
En esta imagen aparece la Raspberry con el IP 192.168.1.20. **Este IP muy probablemente es diferente en cada router.**

Se puede ver que como esta app (Fing) muestra claramente qué dirección IP tiene la Raspberry

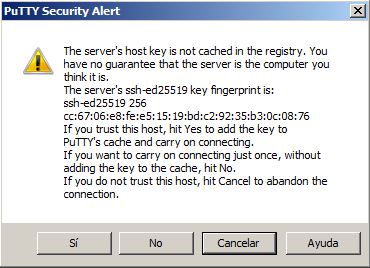


Dirección IP Raspberry

1. Abrir Putty, escribe la dirección IP averiguada antes y clic en **Open**

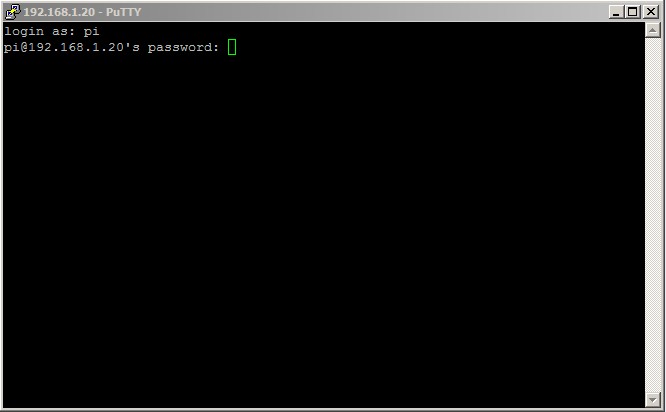


1. Debe salir una ventana de advertencia de seguridad al entrar por SSH por primera vez, clic en **sí**.

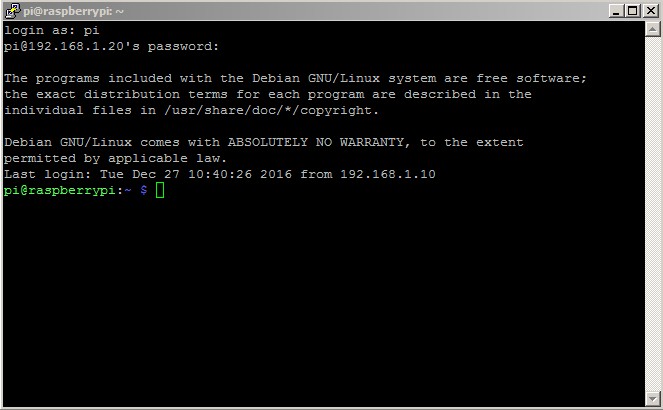


Usuario: **pi**

Contraseña: **raspberry**



Esta es la ventana que se muestra cuando ya hay conexión por SSH con la Raspberry



**Configuración avanzada**

Antes de realizar la configuración avanzada se debe actualizar el sistema

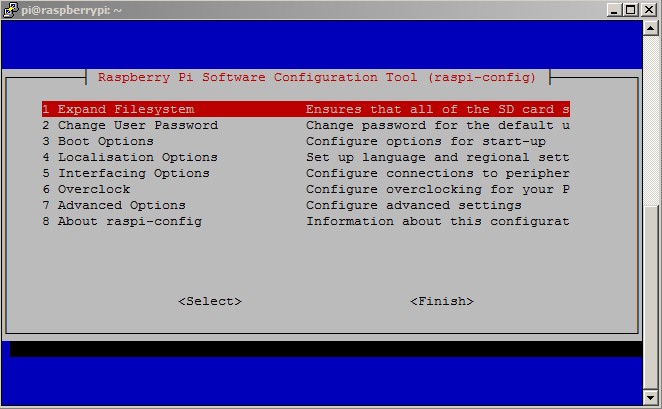
sudo apt-get update

sudo apt-get dist-upgrade -y

Lo siguiente que se debe hacer es completar algunos pasos de configuración avanzada

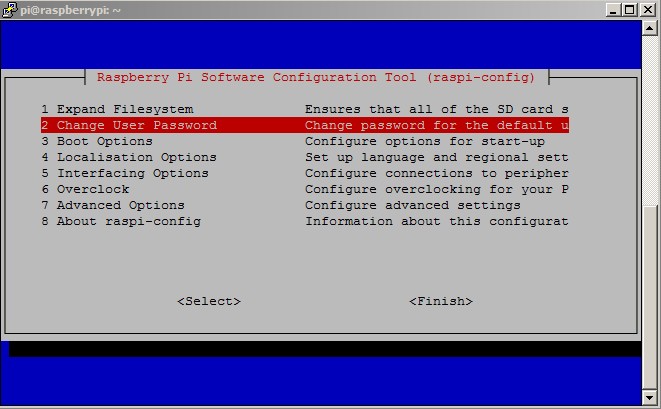
sudo raspi-config

#### “1. Usar el tamaño total de la tarjeta SD”



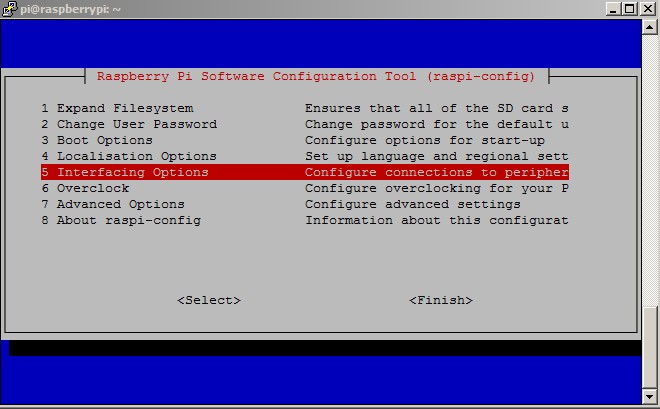
"**1 Expand Filesystem**". Cuando se instala el sistema operativo éste usa el espacio justo en la tarjeta SD para ejecutarse. Al ejecutar esta orden el sistema tendrá disponible todo el espacio de la tarjeta SD.

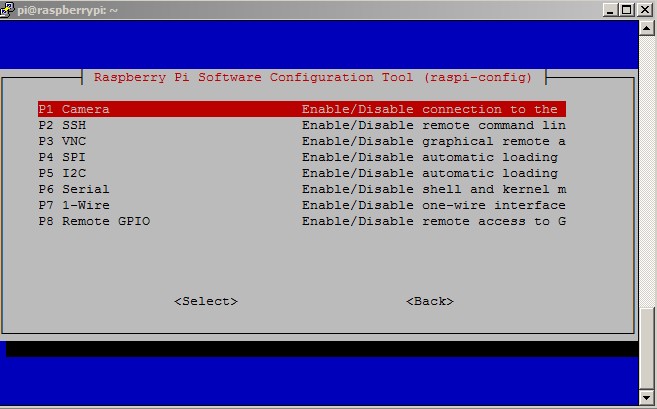
#### 2. Cambiar la contraseña que viene por defecto

****

Es recomendable cambiar la contraseña que viene por defecto con el sistema: "raspberry". Si sólo la se usa para proyectos sin internet, entonces no hay mayor riesgo de seguridad; si se usa con internet, lo recomendable es cambiarla.

#### “5. Opciones de interfaz”





* **Cámara**: Activa/desactiva la cámara de Raspberry **(Raspicam)**. Se trata de la cámara que se conecta al puerto que tiene la Raspberry exclusivo para la cámara, **NO** se trata de cámaras USB. Aunque se desactive esto las cámaras USB **sí** funcionarían.

|  |
| --- |
| http://aprendoycomparto.es/img/camconnector.jpg |
| *Conector exclusivo Raspicam* |

* **SSH**: Activa/desactiva el servicio SSH, permite conectarse a través de una interfaz de texto de modo seguro, con programas como Putty por ejemplo
* **VNC**: Activa/desactiva el servicio VNC, permite conectarte a través de una interfaz gráfica, menos aburrido que el SSH.

1. Reserva de dirección IP.

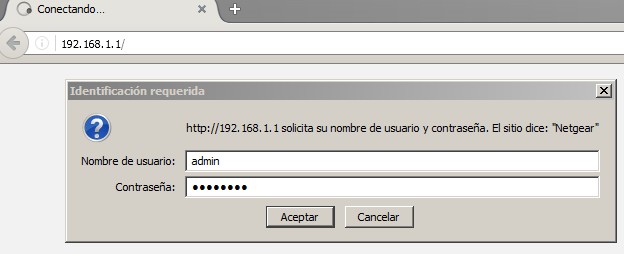
Una reserva de dirección IP se realiza con el propósito de conservar siempre la misma IP en un dispositivo.

**¿Por qué hace falta reservar la dirección IP?** Porque el router tiene un servicio llamado **DHCP** el cual asigna una dirección IP la Raspberry, normalmente una IP aleatoria cada vez que nos conectamos al él. Por ello se hace necesario conservar la misma dirección IP para no tener que buscar qué dirección IP nos ha asignado el servicio DHCP cada vez que reiniciemos la Raspberry.

Para reservar una dirección IP en un router se debe acceder a su configuración a través de un navegador web, aunque en algunos casos tienen una aplicación de configuración.

Hay que acceder al router a través del navegador (Chrome, Mozilla, Internet Explorer, etc.) escribiendo la dirección [http://192.168.0.1](http://192.168.0.1/) o [192.168.1.1](http://192.168.1.1/) en la barra de direcciones **NO en el campo de búsqueda del navegador**. Una de estas dos direcciones debe ser válida para entrar.

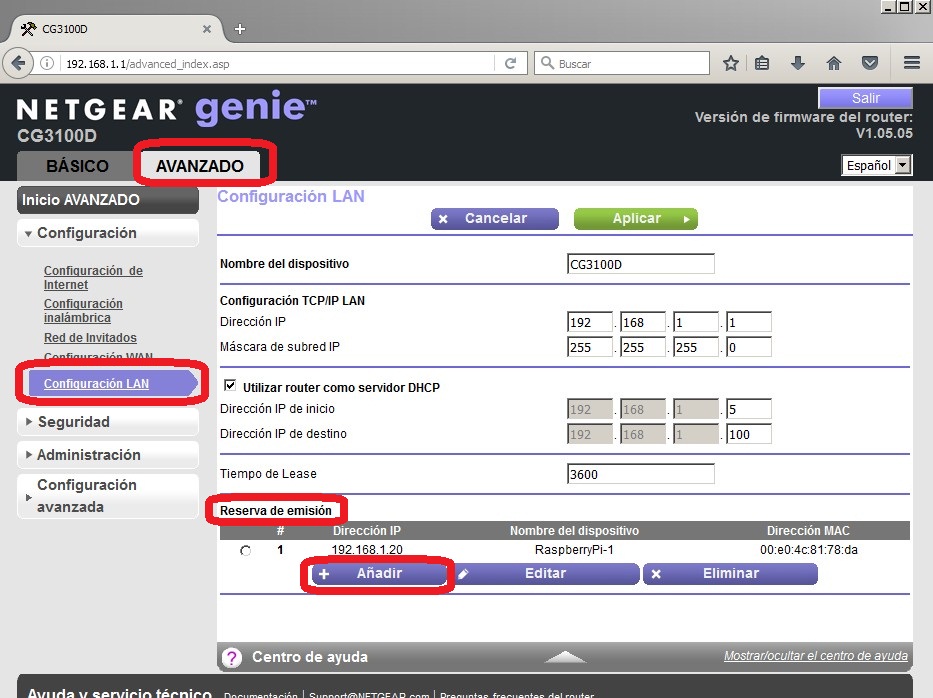
Una vez se intente acceder con una de las direcciones anteriores debe aparecer una ventana pidiendo un nombre de usuario y contraseña, esta información se puede ver físicamente en la parte de atrás o por debajo del router. Ejemplos de algunos usuario/contraseña son: admin/1234, admin/12345, 1234/admin, admin/password, admin/admin

*Ventana de acceso al router*

 Se debe tener en cuenta que cada fabricante define una configuración diferente para sus dispositivos y probablemente la interfaz de acceso varíe de un router a otro.

Buscar la **configuración avanzada**

Busca la configuración LAN, Reserva de emisión (en inglés puede aparecer como **Address reservation**) y vincula la dirección MAC del adaptador inalámbrico con la dirección IP que quieras.



1. Instalación de php en el Raspberry Pi

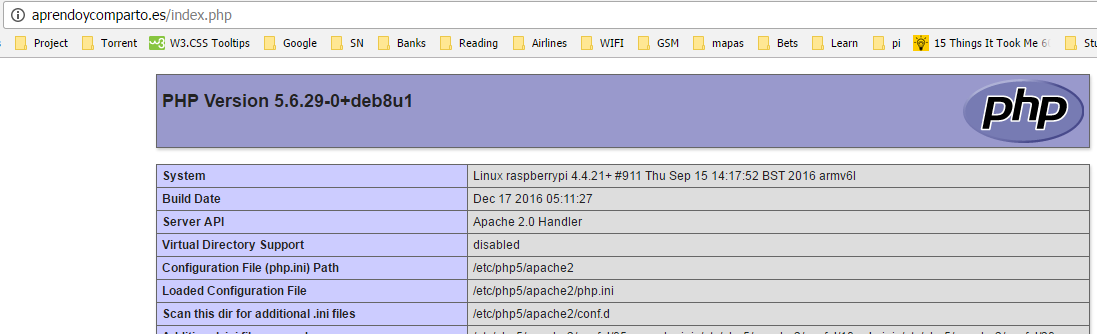
En la última versión del sistema operativo de la Raspberry Raspbian Jessie (abril 2017) PHP ya viene instalado, sin embargo aquí se explica cómo instalarlo

sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache php5-mysql php5-curl php5-gd

Se puede comprobar si la instalación está bien creando un archivo:

sudo echo "<?phpphpinfo(); ?>" | sudo tee /var/www/html/index.php

Ahora en el navegador se comprueba la dirección <http://192.168.1.20/index.php> y debería mostrar algo así:



1. Instalación de Apache en el Raspberry Pi

En la última versión del sistema operativo de la Raspberry Raspbian Jessie (abril 2017) Apache ya viene instalado, sin embargo aquí se explica cómo instalarlo.

sudo apt-get install apache2 apache2-utils